AD-A069 380 BATTELLE COLUMBUS LABS OH F/G 13/13
STRUCTURAL ANALYSIS VIA GENERALIZED INTERACTIVE GRAPHICS -STAGI--ETC(U)
SEP 79 L E HULBERT, N D GHADIALI F33615-76-C-3125

UNCLASSIFIED AFFDL-TR-79-3074-VOL-1 NL

1 OF 1
AD A089 380

END
DATE
FILMED
10-80
DTIC

AFFDL-TR-79-3074
Volume I

AD A089330

STRUCTURAL ANALYSIS VIA GENERALIZED INTERACTIVE GRAPHICS

STAGING

Volume I — Summary Final Report

L. E. HULBERT
N. D. GHADIALI

BATTELLE COLUMBUS LABORATORIES
505 KING AVENUE
COLUMBUS, OHIO 43201

SEPTEMBER 1979

TECHNICAL REPORT AFFDL-TR-79-3074, Volume I
Final Report June 1976 — September 1979

AIR FORCE FLIGHT DYNAMICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

Bernard H. Groomes
Project Engineer

Fred A. Picchioni
Frederick A. Picchioni, Lt Col, USAF
Chief, Analysis & Optimization Branch
Structures & Dynamics Division

FOR THE COMMANDER

Ralph L. Kuster Jr.
Ralph L. Kuster Jr., Col. USAF
Chief, Structures & Dynamics Division

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFWAL/FIBR, *W-PAFB, OH 45433 to help us maintain a current mailing list".*

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

AIR FORCE/56780/29 August 1980 — 380

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
AFFDL-TR-79-3074- Vol-1

2. GOVT ACCESSION NO.
AD-A089 380

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
STRUCTURAL ANALYSIS VIA GENERALIZED INTERACTIVE GRAPHICS - STAGING. VOLUME I. ~~FINAL REPORT~~

5. TYPE OF REPORT & PERIOD COVERED
TECHNICAL - FINAL rept
JUN[redacted]76 - SEP [redacted]79

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
L. E. HULBERT, N. D. GHADIALI

8. CONTRACT OR GRANT NUMBER(s)
F33615-76-C-3125

9. PERFORMING ORGANIZATION NAME AND ADDRESS
BATTELLE, COLUMBUS LABORATORIES
505 KING AVENUE
COLUMBUS, OHIO 43201

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
AIR FORCE FLIGHT DYNAMICS LABORATORY (FBRA)
WRIGHT-PATTERSON AIR FORCE BASE
OHIO 45433

12. REPORT DATE
SEP[redacted]79

13. NUMBER OF PAGES
27

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

INTERACTIVE GRAPHICS	MESH GENERATION
FINITE ELEMENT MODELS	COMPUTER AIDED DESIGN
STRUCTURAL ANALYSIS	COMPUTER AIDED ANALYSIS

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

STAGING (STructural Analysis via Generalized INteractive Graphics) has been developed to give engineers an interactive graphics system for constructing and studying finite element models and for reviewing the results of a finite element analysis.

Volume I gives an overview of the general capabilities of the STAGING system.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

FOREWARD

This final report was prepared by the Columbus Laboratories of Battelle Memorial Institute, Columbus, Ohio, for the Structures and Dynamics Division, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio. The work was performed under Contract No. F-33615-75-C-3125, which was initiated under Project No. 2401, "Structures and Dynamics", Task No. 02, "Design and Analysis Methods for Aerospace Vehicle Structures". Initially, Mr. L. Bernier (FBR) was the AFFDL project engineer for this effort, after which Mr. B.H. Groomes (FBR) was assigned the responsibility.

STAGING, as described in this report, represents a three-year combined Air Force-Navy effort, with specific support and contributions from Dr. Charles P. Poirier, Chief, Scientific Systems Analysis Branch, ASD Computer Center, Wright-Patterson Air Force Base, Ohio, Messers. James M. McKee and Michael E. Golden, Computation Mathematics and Logistics Department, Code 1844, Mr. Paul Mayer and Miss Jane A. Figula, Structures Department, Code 1730.5, The David W. Taylor Naval Ship Research and Development Center, Bethesda, Maryland. The technical graphics expertise of these government researchers are gratefully acknowledged.

The report consists of four volumes. Volume I, "Final Summary Report", presents an overview of the capabilities of the STAGING (STructural Analysis via Generalized INteractive Graphics) system. Volume II, "Users Guide", gives detailed instructions on how to use STAGING for finite element analysis. Volume III, "System Manual", describes the internal structure of STAGING and details procedures for installation and Maintainance of the System on CDC CYBER and 6000 series mainframe computers. Volume IV, "Appendices to the System Manual", includes lists of STAGING procedures, loader directives and cross-referenced tables of all entry names that occur in STAGING.

The program manager of this development was Dr. L. E. Hulbert of the Transportation and Structures Department. He was supported by N. D. Ghadiali of the same department and by a number of specialists from the Computer, Information Systems, and Education Department including:

E. Edwards K. Cadmus

D. Kasik C. Scofield

W. Young F. Drobot

The work reported herein was conducted during the period of June 28, 1976 through June 1979. Some work on STAGING was carried out under contract F33615.

The present report was submitted for publication in June, 1979.

TABLE OF CONTENTS

SECTION I

INTRODUCTION

Through intensive development during the past two decades, finite element analysis has become the predominant method for doing structural analyses of all kinds. Current finite element analysis computer programs can be used to analyze nonlinear geometric deformation and elastic or plastic deformation in any of the parts of extremely complex structures. Many millions of dollars have been invested in developing single computer programs, such as NASA's NASTRAN Program,

Despite this widespread development and use, however, problems remain in applying these computer programs. The biggest single problem is that both the preparation and validation of input data and the interpretation of results are difficult and time consuming. By far the most promising approach to eliminating this problem is the combined use of computer automated mesh generation to minimize data preparation and interactive graphics for data validation. Such a combined computer program has been developed by Battelle's Columbus Laboratories working with the Air Force Flight Dynamics Lab and the David W. Taylor Naval Ship Research and Development Center (DWTNSRDC). This program is called STAGING, an acronym for STructural Analysis via Generalized INteractive Graphics.

The Air Force first considered an interactive graphics program in the early 1970's. About the same time, Battelle and DWTNSRDC were independently considering similar systems. Formal development of the STAGING program was begun by Battelle's Columbus Laboratories under Air Force Flight Dynamics Laboratory Contract Number F-33615-75-C-3096 (April 1975 through April 1976). Shortly thereafter DWTNSRDC joined the STAGING Team. Two STAGING modules - the display and edit module and the data base manager module - were developed in this initial contract.

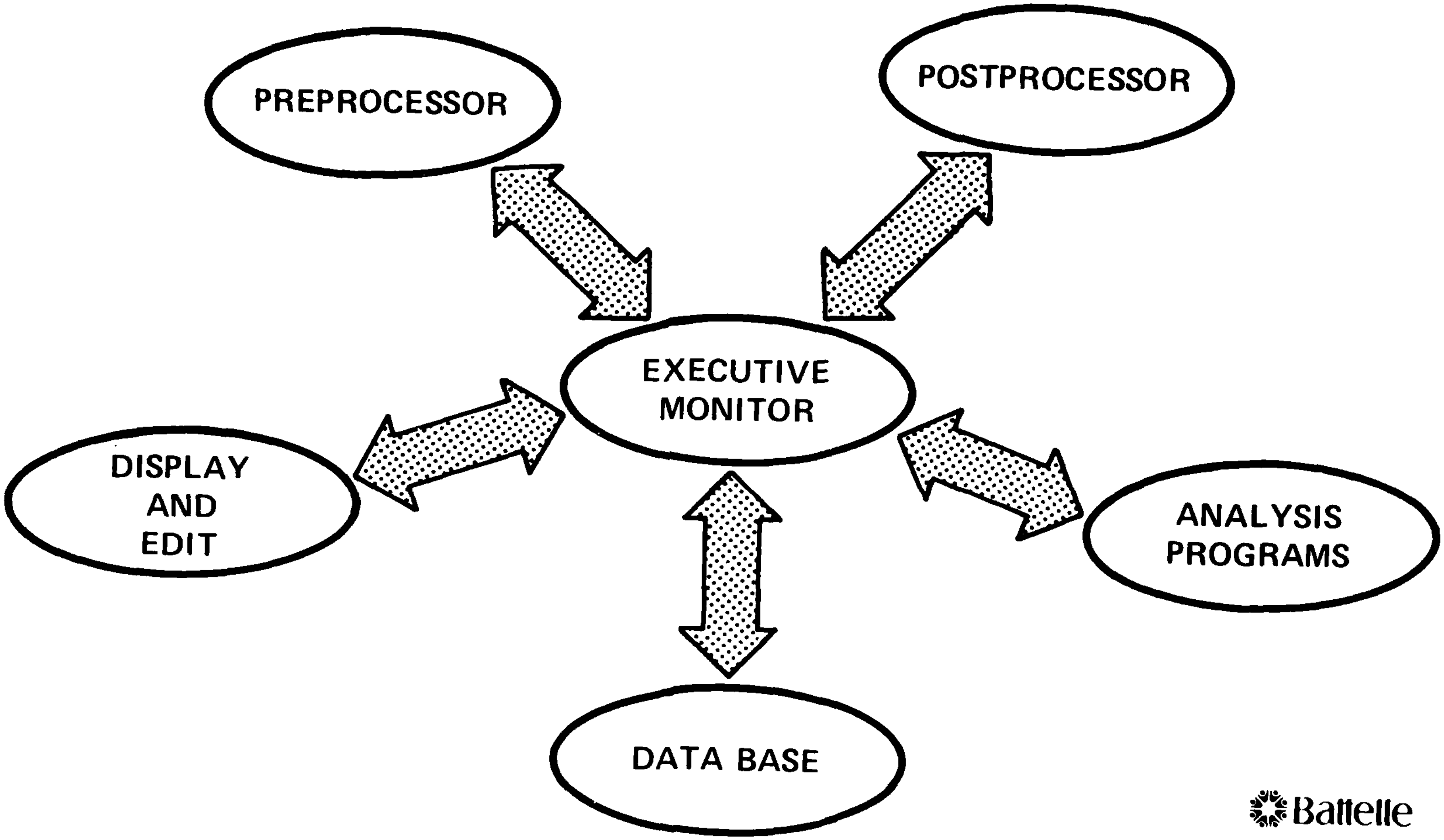
Contract F-33615-76-C-3125 was initiated in June, 1976 to enhance these capabilities by adding further modules to STAGING. This Volume I "Final Summary Report" gives an overview of STAGING capabilities as user's would access them to carry out a finite element analysis. The Volume II

"Users Manual", Volume III "System Manual" and Volume IV "Appendices to the "System Manual" give detailed descriptions of STAGING and its use. Together, these four volumes constitute the final documentation of STAGING as developed under the subject contract. Contributions of DWTNSRDC researchers in developing STAGING's preprocessor system, GPRIME, are described in this report but a GPRIME user's guide is not included.

STAGING runs on Control Data Corporation 6000 and Cyber Computer configurations using Tektronix graphics terminals. It is intended to place structural analysis as much as possible at an engineer's fingertips. The system helps an engineer to generate and validate finite element models and evaluate the results of each finite element analysis. A key feature is the ease with which the experienced user may interface STAGING to any given finite element analysis program. This feature makes STAGING widely applicable since any model input and resultant analysis output may be displayed. STAGING also includes a large number of utility programs that make the handling of pre- and post processed information from analysis of a wide variety of finite element problems as easy and straightforward as possible.

STAGING is a modular system controlled by its own "executive monitor". (cf. Figure 1). The different modules, designed to help the engineer with different tasks in his finite element analysis are:

- o Preprocessor module helps the engineer generate the finite element model.
- o Display and Edit module-helps the engineer to evaluate his model, locate errors, and make needed corrections.
- o Analysis interfacing - provides for user implementation for interfacing any analysis program to STAGING. Includes data conversion routine to convert the model from the STAGING Data Base format to the input data format of the selected finite element program. A data conversion routine also converts the analysis program output results to STAGING Data Base format for interactive display and study once the analysis is finished.
- o Postprocessor-helps the engineer study the results until he fully understands them.



Battelle

FIGURE 1. THE STAGING MODULES

SECTION II

PREPROCESSOR MODULE

In the past, the cost of computing placed a severe limitation on the type of structural model an engineer could analyze. Preparation costs for such models were a small part of the total analytical costs. Now, with the steep drop in computing costs, the engineer can afford to analyze complex models except for the fact that the preparation costs for such models now greatly exceed the computer costs for analyzing the model.

STAGING allows the user to drastically cut these model preparation costs by providing both two-and three-dimensional geometry generation capabilities controlled through the interactive graphics interface. With this interface the STAGING user maintains near real time control of model generation. Moveover, each step of the model building process is displayed graphically for user verification before going on to the next step.

STAGING provides two types of model generation capabilities. One is a two-dimensional generator called DRAFTING. The other is a three-dimensional geometry and mesh generation system called GPRIME. DRAFTING allows the user to create a planar model directly on the graphics screen. He creates this model by first picking the locations of the mesh points with the screen cursor and then creating elements by selecting the points that make up each element. On request, DRAFTING will draw a ruled grid on the graphics terminal to any degree of refinement the user wants so that he can precisely position the node points. (Figure 2). DRAFTING allows the user to either create a model from scratch or add to an existing model in the STAGING Data Base.

The GPRIME geometry generation system was developed at David W. Taylor Naval Ship Research and Development Center by the Computation, Mathematics and Logistics Department. It allows anyone with a rudimentary background in the concepts of geometry to easily define general curves or surfaces in three dimensions. The data for specifying these curves or surfaces may come from many sources, such as points obtained from computer printouts, or points digitized from a blueprint. GPRIME allows direct

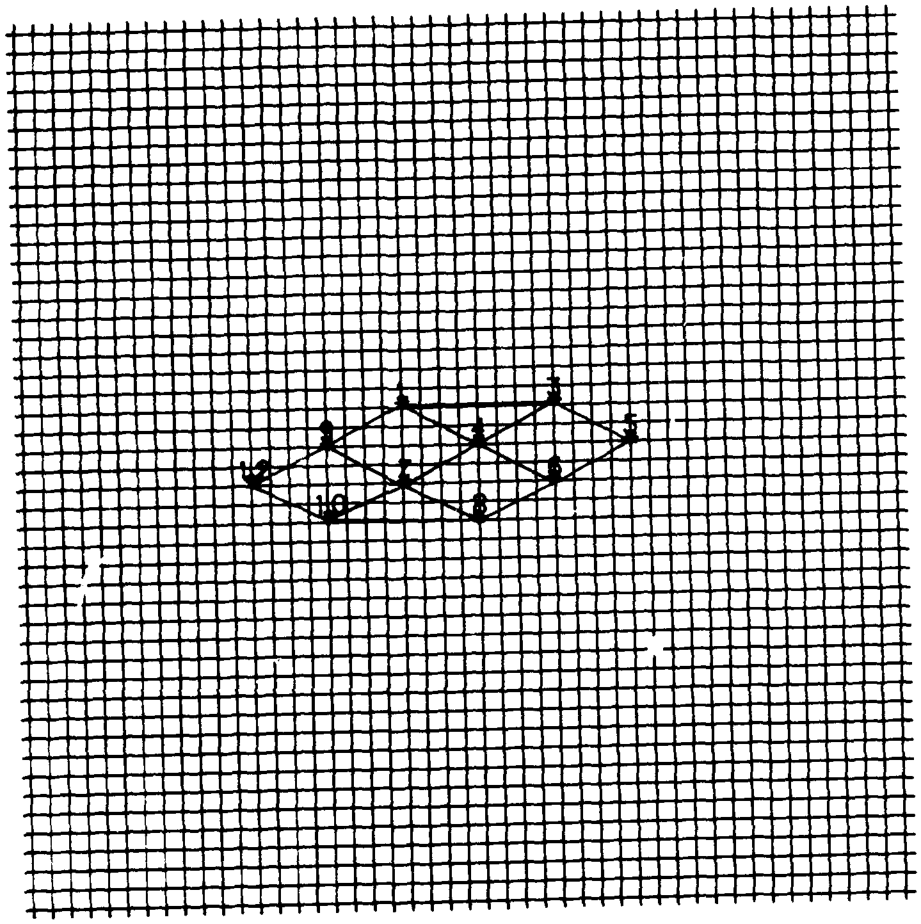


FIGURE 2. A DRAFTING GENERATED PLANE FIGURE

specification of geometric shapes such as cylinders, cones, and spheres using definition statements involving only two or three input parameters. The configurations of most manufactured parts consist of a collection of such simple geometrical shapes.

The GPRIME language includes both definition statements and command statements. Definition statements are used to define a geometric entity in terms of appropriate parameters. A single point is defined in terms of its coordinates; a line in terms of its two endpoints; a circle in terms of its center and radius; and a cylinder in terms of the endpoints of its axis and its radius. GPRIME command statements include operations that are carried out on the geometric entities defined in the GPRIME definition statements. These commands include a variety of viewing options that the user may invoke for displaying the geometry of the structures that he is generating. Options include zooming commands, rotation commands, and hidden line algorithms. GPRIME also includes move and copy algorithms for generating repeated structures. Figure 3 shows one structural shape constructed with a few GPRIME commands.

GPRIME provides a macro command which can itself be defined in terms of a set of GPRIME primitive commands. This macro command allows the user to create a complex structural shape that may be used over and over again. The user may define formal parameters for the macro command that can be used to control the dimensions of the structure. Thus, the user, once he has built this macro command, need only call the command and enter values for the formal parameters to create a new structural shape.

When the user has created a desired structural shape, he can then construct a finite element model directly from this shape using GPRIME commands. GPRIME allows the user to create a three-dimensional shell model by discretizing the created surfaces into shell elements of a size he selects. Alternatively, he can create a three-dimensional finite element model of solid elements bounded by the created surfaces. For complex model shapes GPRIME allows sectioning the model into simpler substructures which can be discretized smoothly into solid or shell elements. As each mesh is generated, GPRIME assembles the substructures until the complete model is created.

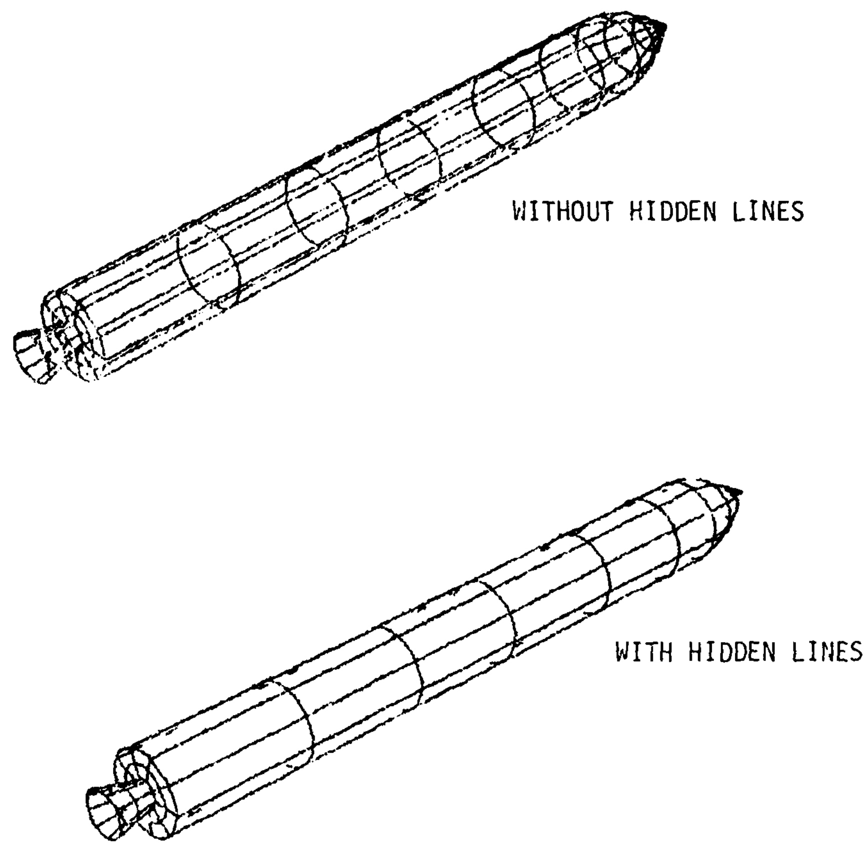


FIGURE 3. A G-PRIME GENERATED SIMPLE MISSLE CONFIGURATION

GPRIME definition and command statements are written in symbolic notation, which helps the user to remember the commands more easily. Moreover, if the user forgets the parameter order required by a particular definition, the GPRIME system has a "help" command to remind him of the necessary input for specifying a command.

The GPRIME approach, then, is to first provide powerful generation algorithms to develop a geometrical model and then directly generate a finite element model from this geometrical model. This significantly reduces the time required to generate finite element models. It also improves the accuracy of the models that are generated. By first seeing the model in three dimensions, the user can better space the elements in critical regions.

In addition to the DRAFTING and GPRIME generators, which are provided with STAGING, interfaces may be written to allow finite element models created by other generators to be stored in the STAGING Data Base. These interfaces are constructed using data conversion routines discussed later. In particular, if the user has a digitizer, he can write interfaces to convert the digitizer output into the STAGING data base for displaying and editing the results. The same is true for data from other mesh generation computer programs. If the user already has a finite element input data deck, data from it can be entered into the STAGING data base with the proper data conversion routines.

SECTION III

DISPLAY AND EDIT MODULE

Once a finite element model has been generated and stored in the STAGING Data Base, the user can interact with this model directly in graphical form. The display and edit module of STAGING gives the user total interactive control over how he wishes to look at his model.

The first step is choosing the part or parts of the model he wants to see. He can display:

- o A single node, or many nodes
- o A single element, or many elements
- o A single substructure, or multiple substructures
- o A complete model.

The user chooses the desired parts by selecting the appropriate items from the command tree "menus", which are discussed in more detail later. He can define any group of elements or nodes as a substructure and define as many substructures as he wishes. Nodes and elements can be shared among any number of substructures.

The user can erase any part of his model from the screen as soon as he finishes studying it. The parts (nodes, elements, substructures) are erased in the same way they are displayed. Thus if the user has activated a set of elements for display, he may erase these elements one at a time or in groups. Conversely if he has activated a substructure for display, erasing this substructure erases all of the elements that were initially drawn as part of it. Of course, erasing parts of the picture does not delete any item from the data base, and the parts can be immediately redrawn if the user wishes to see them again.

In addition to interactively choosing the parts of the model he wishes to see, the user can choose how he wants to look at these parts by using the picture modification commands of STAGING. These commands are part of the "global" set of "menus" that are available to the engineer at all times. These are also discussed in greater detail later.

The picture modifying commands are the most used of all global menus because they allow the user to look at his picture from any kind of

view. "Enlarge" allows the user to zoom in on a selected part of the model that he has displayed. "Rotate" allows him to rotate the model around any of the coordinate axes. "Shrink" allows him to shrink the elements in the system. He can construct a split screen picture allowing him to view from two to four pictures simultaneously. He can ask for a perspective view of the model from any distance and any eye position. (Figure 4)

With the capabilities provided by STAGING, the user has virtually unlimited freedom to study his model graphically until he is satisfied it is free of error. Experience with STAGING has shown that all geometrical errors can be found before an analysis is attempted.

Once an error is detected, it can be immediately and interactively corrected with the Edit portion of STAGING. It has been found that correcting each error interactively as it is found is a faster and surer way to build a correct model than is making several corrections simultaneously in a batch run.

STAGING editing capabilities include.

- o Adding new items to the data base
- o Deleting items from the data base
- o Changing specifications of items in the data base.

An "item" may be either a node, an element, or a substructure. Thus, a new node can be added to the data base by giving it an unused node number and specifying its coordinate position. A new element may be added by giving it an unused element number and by selecting the already defined nodes that belong to the element. Finally, a new substructure can be added by giving it a name and selecting the appropriate nodes and elements needed to construct it. Necessary attributes, such as element type or material type, can be interactively entered as part of the item description. Items already in the data base can easily be deleted either one at a time or in groups. Since STAGING allows the user to work on a copy of his original data base, inadvertently deleting too much of his structure is not catastrophic. He simply recovers either his original, or the data base he saved last and continues his editing task.

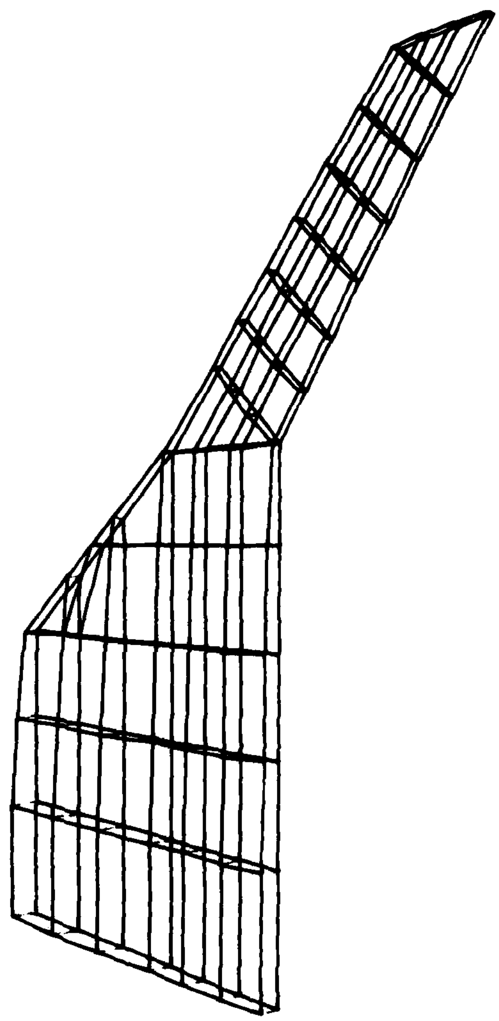


FIGURE 4. PERSPECTIVE PLOT OF ADVANCED DESIGN COMPOSITE WING MODEL.

One of the most common mistakes made in generating a finite element model is to omit a node reference in an element or to mix up nodal references. Errors of this type are easily visible in the graphic display of the model. STAGING provides various convenient ways to directly modify these mistakes. Other non-geometric specifications contained in the attribute lists can also be changed directly.

The substructure concept is important in relieving the user of the need to make tedious, element-by-element, or node-by-node, changes. For example, STAGING allows a set of elements defined as a substructure to be translated as a single unit via the use of the crosshairs. Such bulk changes can be made also in non-geometric data stored in the attribute arrays. Thus, for example, all of the element material specifications in the substructure can be changed in one operation.

With these graphical and nongraphical editing capabilities, STAGING allows the user to arrive quickly at a fully debugged and visually verified finite element model. This model is then converted to the input formats necessary for a finite element program through STAGING'S analysis interface capabilities.

SECTION IV

ANALYSIS INTERFACE

One of the fundamental design objectives of STAGING development was to provide an interface between STAGING and existing finite element analysis programs. Finite element programs are primarily run on batch machines and use card image input and either punched card, printed, or passive graphics output. The purpose of interfacing STAGING with these analysis programs is twofold. First as described above, STAGING is used to check all of the input data to make sure that it is totally correct. Second, once the analysis is run, STAGING allows the user to interactively evaluate the results of the analysis in graphical form.

Communication between an analysis code and STAGING is accomplished through data conversion routines. These routines either transform data from the analysis code formats into STAGING data base format or the reverse. If the user has constructed an input data deck for his finite element model, a data conversion routine reads this input data deck, selects from it those cards that define the geometry of the model,and stores the data in a STAGING Data Base. As described above, the user then studies his model and uses the STAGING display and edit module to make the appropriate corrections. With all the corrections made, the user constructs a corrected input data deck for the finite element analysis. To accomplish this, a second data conversion routine takes the corrected geometric data from the STAGING data base and integrates it with the rest of the input data for the user's program.

Once the analysis has been completed, results are stored in the STAGING Data Base so that the user can graphically display them, study them, and prepare hard copies for his reporting needs. For this, a third data conversion routine captures the results output and adds the data to the user's existing model data base.

Since the data conversion routines involve mapping the input and output data of the analysis programs into and out of the STAGING data base, the size of the data conversion routines clearly depends on the size of the input/output data routines used in an analysis code. For simple

analysis codes, simple data conversion routines are needed. For general purpose analysis programs, such as NASTRAN, FASTOP, ADENA, or MARC, which have complex input/output data requirements, the data conversion routines must be much more complex. For such programs, it is often desirable to develop data conversion routines for the nodes, elements, and output results for load cases that the user needs most often. Such data conversion routines are quickly developed and can be easily generalized for new elements or load cases when needed.

Data conversion routines have been generated and incorporated with STAGING for the following programs:

a)	two-dimensional	AXISOL, DOASIS, HONDO
b)	three-dimensional	NASTRAN, ADINA
c)	optimization	FASTOP.

Descriptions of these conversion routines are given in the STAGING User Manual, Volume II of this report.

As mentioned in the preprocessor discussion, data conversion routines can be developed easily to provide interfaces between STAGING and such model generators as digitizers or stand-alone mesh generators. These data conversion routines capture the output of the data generators (which are usually in the form of node coordinates and element nodal connection tables) and convert these data to the STAGING data base. Note that once a data base is created, no matter by what means, all of the STAGING facilities are at the user's disposal to study the model, correct it, and use it to prepare the input data for the analysis program he has selected.

To summarize, STAGING provides data conversion routines for communication between analysis programs and the STAGING data base. This approach gives the user great flexibility in choosing the particular analysis program that he wants to use and in choosing the most convenient way of generating the input data for that analysis program.

SECTION V

POSTPROCESSOR MODULE

As noted in the previous section, STAGING allows structural analysis results to be entered in the data base, using data conversion routines. These results must be added to the original data base that defines the structural model. Each item of the results is stored as an attribute of the node or element for which it is computed. Thus, nodal displacements are entered in the data base as nodal attributes. Elemental stresses are stored as elemental attributes.

STAGING has the flexibility to store and display almost any kind of result that is computed by an analysis code. However, with STAGING the user may display results specifically related to finite element analysis. These include deformed shapes and stress contour plots. The user may also automatically draw two dimensional x-y plots showing variations in his output results. Any of the results may be plotted against any parameter of the user's choosing.

Plots of the deformed shape of any structure can be automatically generated by STAGING from the original structural shape and the computed nodal displacements. Displacements can be the result of a single load condition or mode shape. Multiple displacement sets resulting from non-linear analyses or dynamic analyses may also be displayed. Scale factors for the displacements can be interactively entered by the user. For multiple plots, the user can interactively select the time step he wishes to see or he can ask for sequential viewing of the deformed structure at one time step after another. The deformed structure can either be plotted by itself or superimposed on the original shape. A dashed line plot of the deformed shape is drawn to distinguish it from the original shape. Figure 5 shows a superimposed undeformed and deformed structure.

Note that after drawing the deformed and/or undeformed structures, the user can call on all of the picture modification capabilities (zoom, rotation, etc.) to study his deformed/undeformed structure in any way he wishes until he completely understands what has happened to the structure.

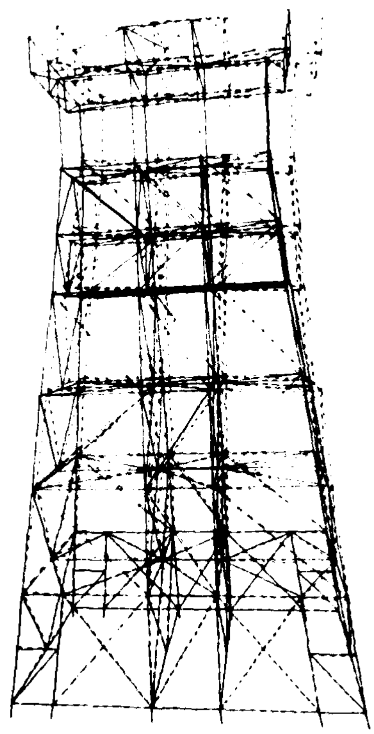


FIGURE 5. DEFORMED PLOT

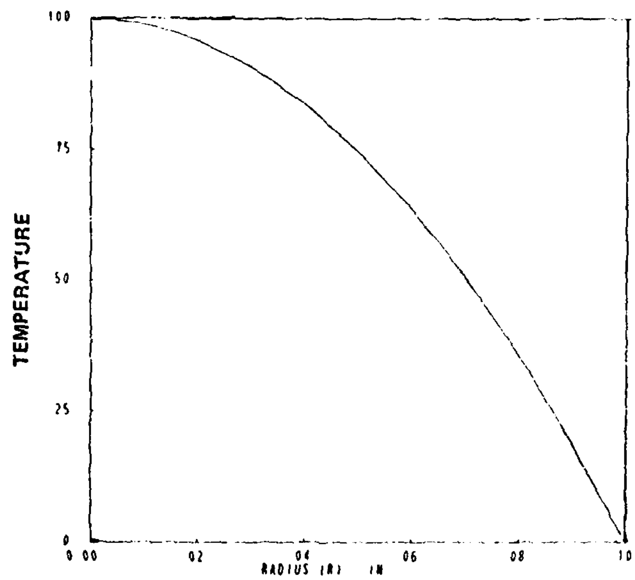
Contour plots can be generated by STAGING for any collection of elements or nodes. The contour lines are always projected onto a principal plane (such as the x-y plane or y-z plane). The engineer can select the number of contour lines and whether or not they are to be labeled.

STAGING allows complete flexibility to plot any attribute in the data base as a function of any other attribute defined for the same set of elements or nodes. To use the x-y plot capability, the user first activates the set of nodes or elements containing the data he wishes to plot. He then designates which attribute should be the plot ordinate and which the abscissa. STAGING automatically plots the curve according to the graph style that the user selects.

The construction of the x-y plot allows the choice of up to ten attribute pairs for plotting. The first plot controls the size and limits of the plotting space. Subsequent curves are plotted in the same style and scale as the first. The first curve is drawn automatically, while the user must trigger the drawing of subsequent curves.

Split screen and zooming capabilities are available as in plotting a two-dimensional model. Further capabilities allow rescaling the plot to fit the largest values within the grid boundaries for all curves displayed simultaneously. Figure 6 illustrates a typical plot.

A particular data transformation that is unique to a given program can be included in the data conversion routine. Thus, the postprocessor module itself can be reserved for the types of transformations that may be commonly used or needed by a number of different analysis programs. The approach used in the postprocessor is to start with the data generated by the analysis module as it is already stored in the STAGING Data Base. Under the control of the Postprocessor command tree, the user carries out the steps needed to generate the data he wants. These data are also stored in attribute arrays in the data base. In the same way, in an analysis program data conversion routine, the user would compute new data arrays and store these, together with the original program output data arrays, as separate attributes in the data base. The postprocessor module can create two new items - principal stresses and equivalent stresses. At the user's request, these are



TEMPERATURE DISTRIBUTION FOR SOLID DISK

FIGURE 6. TYPICAL X-Y PLOT

generated from the normal and shear stress output of the user's analysis program and stored in the STAGING data base for display and study.

SECTION VI

STAGING UTILITIES

Much of a user's communication with a computing system, such as STAGING, is in the form of a set of commands to trigger computer actions. Relatively simple computer programs have a small set of commands. It is very easy for the user to learn these commands and any restrictions in the way they are used. However, with a system as complex as STAGING, it is virtually impossible for a user to memorize all of the required commands. Yet, since STAGING is an interactive system. the user must be able to enter a sequence of commands in rapid succession in order to use the system. and his own time, efficiently.

STAGING attacks this problem by using a command tree approach. In this approach, the commands used by STAGING are structured into a logical tree. The user traverses this tree in a structured manner to carry out his STAGING session. At each level of the tree, a set of STAGING commands is presented to the user in the form of a menu. Only valid choices in the menu are presented at any given time, freeing the user from any concern about the validity of command choices. Once the user chooses a command from the menus, one of two things can happen. Either the next level of STAGING commands is presented to the user or STAGING takes action as required by that STAGING command. The use of a structured command tree literally leads the user through STAGING and relieves him of the burden of memorizing what command should come next. This greatly speeds the learning process for the user, since he rapidly learns where in STAGING a given type of activity is carried out.

In addition to the general command tree facility STAGING also contains a set of globally available commands. The STAGING Global commands relate to a number of activities (such as picture modifying commands discussed earlier) that the user may want to invoke from different parts of STAGING. Other Global utilities include the capability to stop the STAGING session at any time. With this "stop", the user can choose whether he wants to save the data base he has created. Another feature allows the user to see a set of statistics about his display or about

his data base at any time. Display statistics include the items that are active in the display and the maximum and minimum dimension of the picture. Data base statistics include lists of items in the data base (structures, substructures, elements, nodes, attributes) and the number of nodes and elements.

SECTION VII

CONCLUSIONS AND ACKNOWLEDGEMENTS

A primary objective in developing STAGING was to significantly enhance the engineer's ability to generate finite element models of complex structures. An equally important goal was to give the engineer a tool that would be easy to use. It was obvious to the original developers that attaining these twin goals required an effort that would overtax the resources to any one organization.

STAGING, as described in this report, represents the combined efforts of the Air Force Flight Dynamics Laboratory, The David W. Taylor Naval Ship Research and Development Center, and the Columbus Laboratories of Battelle Memorial Institute.

STAGING does enhance the engineer's productivity and is easy to use. It is thus proof that the three organizataions could and did work together effectively. In addition to the current contract (F-33615-76-C-3125), STAGING work at BCL was supported by an earlier AFFDL contract (F-33615-75-C-3096). This support and the contributions of the researchers at David Taylor to the development of the current STAGING System are gratefully acknowledged. They were both essential to the success of STAGING development.

☆U.S.Government Printing Office: 1980 — 657-084/71